Remarks

Applicants respectfully request reconsideration of the application in view of the foregoing amendments and following remarks. With entry of the present amendment, claims 1-20 remain pending.

Patentability Over Cited Art

Claims 1, 3, 4, 6-8, 10-12 and 20 have been rejected under 35 U.S.C. § 103(a) as allegedly being obvious over Sloan, in view of Burke and Purcell. Claims 2, 5, 12, 13 and 15-19 have been rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over Sloan, in view of Morioka, Burke, and Purcell. Applicants traverse the rejection.

Sloan, Morioka, Burke and Purcell lack recited textures containing data values representing normals and positions of sample points.

The independent claims each contain language relating to creating texture data structures containing positions and normals of a set of points sample over an object whose image is being rendered, and then performing a set of texture-based operations on the positions and normals textures using a graphics processing unit.

In particular, claim 1 recites, "creating an object positions texture containing a set of data values representing positions of a set of points sampled over the object mapped into a texture space," and "creating an object normals texture containing a set of data values representing normals of the set of sampled points mapped into the texture space." The language is amended to more clearly emphasize that the positions and normals are represented by data values contained in the textures. Claim 1 further recites actions of "determing cosine terms," "determining shadowing," and "determining radiance transfer contribution" each performed "as a texture-based operation using the graphics processing unit." The other independent claims 2 and 3 likewise recite the normals and positions are contained in textures, which textures are then processed in texture-based operations of a graphics processing unit.

The Office recognizes that "Sloan fails to teach creating an object positions and normal texture," and asserts "Burke teaches creating an object positions texture representing positions of a set of points sample over the object… and object normals texture representing normals…" at paragraph 0035, lines 4-10 of Burke. Applicants respectfully disagree. Burke lacks any teaching

or suggestion to put position and normals data values into textures for processing using texture-based operations of a graphics processing unit, and then performing texture-based operations using a graphics processing unit to process such positions and normals data in the textures. At the cited paragraph 0035, lines 4-10, Burke indicates each sampled point is represented by nine data values, position (X, Y, Z), color (R, G, B), and normal (I, J, K). Burke also describes a model data creator creates a data structure for this nine-dimension data values of the sampled points, and indicates the data structure is "in the form of a network where each point has four pointers [indicating next points in U, V directions]." *See*, Burke at paragraph 0036. Such data structures in the form of a network of pointers is not a texture that can be processed using texture-based shader operations on a graphics processing unit. Because Burke describes creating a data structure <u>in the form of a network of pointers</u> for this nine dimension position, color and normal data of sampled points, Burke fails to teach or suggest to create textures for processing using texture-based operations of a graphics processing unit that contain data values representing position and normals as recited in the present claims.

<u>Sloan, Morioka, Burke and Purcell lack radiance transfer computation with outer loop iterating over directions, and inner loop iterating over points.</u>

The independent claims 1, 2 and 3 each recites language relating to a radiance transfer computation with an outer loop iterating over directions, and inner loop iterating over points. For example, claim 1 recites, "iteratively, for each of a set of directions sampled about the object," performing various texture-based operations ("determining cosine terms," "determining shadowing," and "determining radiance transfer contribution") over a set of sampled points. Claims 2 and 3 recite like language.

Claim 1, as amended, further recites the claimed method "produc[es] a radiance transfer value for each of the sampled points from the accumulated radiance transfer contributions for the iterated directions at the respective sampled points." Claims 2 and 3 are amended to recite like language.

The calculation of radiance transfer for a set of sampled points using an outer loop iterating over directions, and texture-based operations on sets of sampled points inside the loop is not taught or suggested by this art.

The Office asserts that it would have been obvious to reverse the inner and outer loops of the method illustrated in Figure 3 of the Specification, in view of the description by Percell of an

optimization to minimize the total number of passes by a ray tracer. Applicants respectfully disagree.

First, mere reversal of the parameters iterated over by outer and inner loops of the method illustrated in Figure 3 would not result in the claimed method that produces radiance transfer at each point which is the accumulated radiance transfer contribution over all directions for that point. For example, the prior art computation as shown in Figure 3 of the present application is as follows:

```
For each point P
     Accum = 0
     For each direction D
          Hn = dot(D,N)
          if (Hn < 0) continue;
          if (RayDoesNotIntersect(P,D))
               Accum += B(D)*Hn
     End For
     T = Accum/Norm
End For
```

If we merely switch the values iterated by inner and outer loops of this process, the result would be:

```
For each direction D
     Accum = 0
     For each point P
          Hn = dot(D,N)
          if (Hn < 0) continue;
          if (RayDoesNotIntersect(P,D))
               Accum += B(D)*Hn
     End For
     T = Accum/Norm
End For
```

This reversed calculation produces a transfer value per direction that is the accumulated contribution from all points for that direction. The transfer per direction produced from the reversed calculation is not the same as the radiance transfer per point that is the accumulated contribution from all directions as recited in claims 1-3. So, merely reversing what value is iterated in the outer and inner loops actually does not produce the radiance transfer at each point for the iterated directions, as recited in the claims.

Second, the cited art would not motivate reversing the values iterated by outer and inner loops of the radiance transfer computation illustrated in Figure 3 of the specification. The Office

alleges that Purcell's description of optimizing a ray tracing procedure "to minimize the total number of passes" would motivate modifying the radiance transfer computation illustrated in Figure 3 of the Specification. Applicants respectfully disagree.

Unlike the ray tracing process described in Purcell, switching the values iterated by outer and inner loops of the radiance transfer process illustrated in Figure 3 of the Specification does not permit optimizing to "minimize the total number of passes" as achieved by Purcell. In Purcell's ray tracer, the outer "while" loop iterates through "rays." *See*, Purcell at section 3.2, ¶ 4. This while loop includes a test of whether any rays are "active." *See*, Purcell at section 3.2, ¶ 5. The "inactive" rays have "either hit triangles or traversed the entire grid," and therefore do not require a further traverse or intersect pass. *See*, Purcell at section 3.2, ¶ 5. The simple procedure, which Purcell lists in section 3.2 between ¶¶ 4 and 5, runs the intersection test if any rays require intersection. *See*, Purcell at section 3.2, ¶ 5. Purcell describes that this procedure can be optimized by deciding to perform "intersection" passes only when 20% of the rays require intersection tests. *See*, Purcell at section 3.2, ¶ 5. By contrast, the radiance transfer method illustrated in Figure 3 of the Specification does not involve such "intersection" and "traversal" passes within its inner loop. Accordingly, the motivation for Purcell's optimization of nesting loops to minimize such "intersection" passes does not apply to the radiance transfer computation in Figure 3.

In fact, in the method recited in claims 1-3, the texture-based operations of "determining cosine terms," "determining shadowing," and "determining radiance transfer contribution" are recited as being preformed "for each of a set of directions." The choice of value iterated by the outer loops does not permit eliminating "passes" performed in the inner loop. Rather, all these steps are performed for each direction iterated in the outer loop. The motivation for the optimization described by Purcell therefore would not have led to changing which value is iterated by the outer loop of the radiance transfer method illustrated in Figure 3.

For at least these reasons, the cited art fails to teach or suggest the limitations recited in the independent claims 1-3. Claims 4-20 are dependent claims, and therefore also not taught or suggested by this art for at least the reasons presented above for their respective base independent claim.

Conclusion

The application should now be in condition for allowance. Such action is respectfully solicited.

Respectfully submitted,

KLARQUIST SPARKMAN, LLP

One World Trade Center, Suite 1600
121 S.W. Salmon Street
Portland, Oregon 97204
Telephone: (503) 595-5300
Facsimile: (503) 595-5301

By Stephen A. Wight
Stephen A. Wight
Registration No. 37,759